Contact Determination for Real-time Haptic Interaction in 3D Modeling, Editing and Painting*

Ming C. Lin Arthur Gregory Stephen Ehmann Stephan Gottschalk Russ Taylor
Department of Computer Science
University of North Carolina
Chapel Hill, NC 27599-3175
{lin,gregory,ehmann,stefan,taylorr}@cs.unc.edu

1 Introduction

Collision detection and distance computation are important for a number of engineering applications including dynamic simulation, tolerance verification, object manipulation, motion planning and control. Numerous algorithms and techniques have been proposed. (See a recent survey [LG98].) In order to meet the stringent requirement of haptic rendering, new algorithms and specialized system implementation need to be developed to substain KHz haptic update rates on complex models. This requires improving the state of the art in contact determination by at least an order of magnitude.

In this paper, we present a general and extensible algorithmic framework for fast and accurate contact determination for haptic display of complex geometric models. Our ultimate goal is to support a wide range of force feedback devices. Given a model, we pre-compute a hybrid hierarchical representation, utilizing both spatial partitioning and bounding volume hierarchy. At run time, we use hybrid hierarchical representations and exploit frame-to-frame coherence for fast proximity queries. We further discuss technical issues involved and propose approaches to improve the overall system performance. An initial prototype system has been implemented and interfaced with a 3-dof PHANToM arm and its haptic toolkit, GHOST, and applied to a number of models. As compared to the commercial implementation, we are able to achieve up to 20 times speedup in our experiments and sustain update rates over 1000Hz on a 400MHz Pentium II.

Based on our prototype implementation "H-Collide", we develop an intuitive 3D interface for interactively editing and painting a 3D polygonal mesh using a 3-dof PHANToM. An artist or a designer can use this system to create a multi-resolution polygonal mesh, further refine it by multi-resolution modeling techniques or enhance its look by painting colors and textures on it. The system allows the user to naturally create complex forms by a sense of touch and to freely interact with the design space without specification of rigorous mathematics.

2 Our Approaches

In this section, we briefly examine the requirements of general haptic rendering systems and then highlight our approaches to the problem of contact determination for haptic display.

*Supported in part by NSF grants EIA-9806027, DMI-9900157, and IIS-9821067 and Intel.

2.1 System Requirements

Any contact determination algorithm or system for haptic rendering should be able to address the following requirements for smooth interaction in virtual prototyping and engineering design: (1) scalable performance on large complex model, (2) unstructured models that may deform over time due to forces or manipulation, (3) multiple contacts and close proximity, (4) extensibility and generality.

These requirements imply that the algorithms need to have the following characteristics: (1) the runtime should be independent of the model complexity, (2) it should work well on dynamic scenes, in addition to static environments, (3) it should be able to handle contacts rapidly and *accurately* to generate realistic forces for haptic interaction, (4) the data structure and the algorithmic framework should be appicable to a wide range of haptic devices and applications.

2.2 An Extensible Design Framework

Based on the problem characteristics and the system requirements, we propose a general design framework, which specializes many earlier algorithms for haptic interaction in virtual prototyping of mechanical and structural design. This framework utilizes:

- **Spatial Decomposition:** It decomposes the workspace into regions (e.g. coarse-grain uniform grid cells, adaptive grids, etc.), implemented as a hash table to efficiently deal with large storage requirements. At runtime, the algorithm quickly finds the region(s) containing the volume swept out by the probe or the bounding volumes of the moving objects, and thereby the "region(s) of potential contacts".
- **Adaptive, Embedded Bounding Volume Hierarchy:** For each region containing some primitives of the objects in the simulated environment, we pre-compute a corresponding bounding volume hierarchy (based on OBB's, SSV's [LGLM99], K-DoP's, Spherical Shells, or others) for that region and store the pointer to the associated bounding volume hierarchy using a hash table for performing constant-time proximity queries. Each hierarchy is an embedded node of the bounding volume hierarchy of the entire model. At run-time, most of the computation time is spent in finding collisions between a bounding volume and the path swept out by the tip of the probe or between a pair of bounding volumes. To optimize this query, we have developed specialized and fast overlap tests that take very few arithmetic operations. This embedded hierarchical representation is adaptively modified for deformations due to external forces or haptic manipulation.
- **Temporal and Spatial Coherence:** The algorithm exploits temporal and spatial coherence by caching the contact geometry from the previous step to perform incremental computations.

After pre-processing, the on-line computation consists of three phases. In the first phase, it identifies "the region(s) of potential contacts" by determining which region(s) are touched by the probe path or the bounding volumes of the objects, using the precomputed look-up table. This allows the algorithm to quickly eliminate many levels of tree traversal by zooming in directly to the portions of subtrees that correspond to the regions of close proximity. In the second phase, it traverses down the bounding volume hierarchies using associated nodes of the region(s) of potential contacts, to rapidly determine if collisions have occurred using the specialized fast overlap test. In the third phase, if the contacts occur, it computes the (projected) surface contact point(s). If contacts occurred in the previous frame, we exploit temporal and spatial coherence by caching the previous pair of contact witnesses to initialize the queries and computation.

2.3 Other Optimizations

In addition to the overall framework, we also need to conduct further investigations to extend and generalize this design framework for more challenging scenarios, including handling surface-surface contacts at KHz rate. There are several algorithmic issues that remain to be addressed: (a) computation of optimal hierarchies, (b) intelligent choice of bounding volumes, (c) specialized overlap tests for spline surface primitives, and (d) more efficient handling of deformable models. Due to the space limitation, we will not go into details about each issue.

3 Implementation and Results

We are currently working on extending the design framework with a 6-DoF haptic device. However, we have implemented many of the algorithms described and developed a prototype system using a 3-DoF force feedback arm. In this section, we briefly describe the initial results we have achieved to indicate the potential of the proposed approaches.

3.1 Prototype Implementation Using a 3-DoF PHANToM Arm

Using on the design framework described in Section 2.2, we have implemented a preliminary version of the algorithms described earlier. For comparison, we have implemented adaptive grids, our hybrid approach and an algorithm using only OBBTrees and the specialized overlap test. We have applied them to a wide range of models of varying sizes (from 5,000 polygons to over 80,000 polygons as shown at http://www.cs.unc.edu/~geom/HCollide/model.pdf). Their performance varies based on the models, the configuration of the probe relative to the model, and machine configuration (e.g. cache and memory size). Our hybrid approach results in a factor of 2-20 speed improvement as compared to a native *GHOST* method.

3.2 Further Enhancement

In addition to the early prototyping system based on our design framework, H-Collide, we also investigated some of the technical issues addressed in Section 2.

Hierarchy Construction: We have implemented a combinational hierarchy construction scheme that uses both the top-down splitting and "tiling" of the polygons. In our implementation, we observed a significant speed up (more than two order of magnitude) when using hierarchies of spheres. However, we did not observe similar performance gain for OBBTrees.

Adaptive, Hybrid Hierarchies: We have implemented a software framework for performing contact determination based on hybrid hierarchies consisting of a family of *swept sphere volumes* [LGLM99]. The desired BV types are specified either at run time or computed statically offline. We observe some modest performance gain only in some cases and have not been able to reach any conclusion regarding the appropriate selection mechanism.

Specialized Overlap Tests: We also have implemented a specialized overlap test between two OBB's with SIMD instruction sets. We were able to obtain an average speed-up factor of 2-3. We plan to implement a specialized overlap test between two higher-order bounding volumes for splines/NURBS models. We believe that a SIMD or mini-parallel implementation can provide similar performance gain as well.

Local Deformation: The adaptive hybrid hierarchy was able to handle local deformation, while substaining the KHz update rate. Based on our prototype system implementation of *H-COLLIDE* [GLGT99], we developed an interactive multiresolution modeling and 3D painting system using a haptic interface, called *inTouch* [GEL99], which we will describe next.

3.3 inTouch

inTouch is an interactive multiresolution modeling and 3D painting system with a haptic interface. An artist or a designer can use *inTouch* to create and refine a three-dimensional multiresolution polygonal mesh. Its appearance can be further enhanced by directly painting onto its surface. The system allows users to naturally create complex forms and patterns not only aided by visual feedback but also by their sense of touch.

We chose subdivision surfaces as the underlying geometric representation for our system. This representation enables the user to perform global shape design and multiresolution editing with ease, allows the users to trade off fidelity for speed, and operates on simple triangular meshes. In addition, our system also offers 3D painting capability on arbitrary polygonal meshes with the haptic stylus as an "electronic paintbrush". The contact information output by H-Collide is used for both model editing and painting.

To deform and shape the model interactively, the user simply chooses the edit level (resolution) and attaches the probe to the surface. The real-time haptic display is rendered using *GHOST* and H-Collide. The deformation update process uses the force vector currently being displayed by the PHANToM to move the current surface point at the selected edit level. These geometric changes are then propagated up according to subdivision rules to the highest level of the mesh. The changes are sent across the network to the client application which maintains an identical multiresolution data structure so that it can perform the same operation to update the graphical display. Once the highest level mesh has been modified, the H-Collide and graphical data structures need to be updated to reflect the change. A local deformation algorithm is used to merge the changed triangles with the triangles that were not changed in the H-Collide data structure. The graphical output subsystem also receives the update and proceeds to modify the display lists corresponding to the changed triangles and redraw the screen.

As for 3D painting, H-Collide is used to establish the contact point of the probe with the surface of the object. The probe is then used as a virtual paintbrush with the user's preferred brush size, color, and falloff. The brush size is stretched relative to the amount of force being applied by the stylus, in a manner similar to real painting. Please refer to [GEL99] for more details about the design and implementation of *inTouch*.

4 Ongoing and Future Work

We are currently working on extending the design framework to support a 6-DoF PHANToM 1.5 to manipulate CAD models, nano-structures and flexible surfaces that may deform due to manipulation. Our ultimate goal is to support haptic interaction with complex CAD models for virtual prototyping and engineering design. We plan to continue extending our current algorithmic framework to general haptic devices and to design new hierarchy construction methods for allowing even faster local modification of surfaces, and to work on seamless integration of algorithmic techniques and data structures.

References

[GEL99] A. Gregory, S. Ehmann, and M. C. Lin. *inTouch*: Interactive Multiresolution Modeling and 3D Painting with a Haptic Interface. Technical report, Department of Computer Science, University of North Carolina, 1999.

[GLGT99] A. Gregory, M. Lin, S. Gottschalk, and R. Taylor. H-Collide: A Framework for Fast and Accurate Collision Detection for Haptic Interaction. In *Proceedings of Virtual Reality Conference 1999*, 1999.

[LGLM99] E. Larsen, S. Gottschalk, M. Lin, and D. Manocha", Fast Proximity Queries with Swept Sphere Volumes. Department of Computer Science, University of North Carolina, *Tech. Report #TR99-018*, 1999

[LG98] M. Lin and S. Gottschalk. Collision Detection between Geometric Models: A Survey. In *Proc. of IMA Conference on Mathematics of Surfaces*, 1998.